AN ARGUS SPECIALIST PUBLICATION

DECEMBER 1985

A&B COMPUTING

£1.50

FOR USERS OF THE BBC MICRO AND ELECTRON

SIMULATIONS
THE SUBURBAN FOX
AND THE FAST CAR

ROM SOFTWARE FOR BEGINNERS

MEMORY MODULE: 64K PERMANENT RAM

UNIX FOR BBC BUSINESS PRODUCTIVITY

SLEUTH II

ALTRA ROMs

FAST FLOOD FILLS

PRINTER BUFFER FOR MODEL B+

BUMPER PACK OF CHRISTMAS GAMES – PLAY SCROOGE!

16K SIDEWAYS RAM FOR £10

SPORTING ACTION: OCEAN'S MATCHDAY FOOTBALL

Volume Three Number

Editor: Mark Webb
Assistant Editor: Fiona J.Eldridge
Group Editor: Dave Bradshaw
Group Managing Editor: Wendy J.Palmer
Advertisement Manager: Jonathan McGarry
Editorial Director: Ron Moulton
Managing Director: Peter Welham

REGULARS

Articles in the next month column are in an advanced state of preparation but cannot be guaranteed to appear.

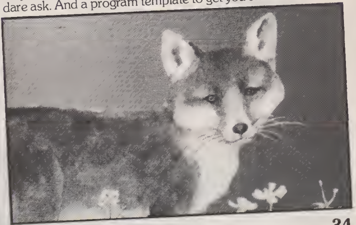
A&B Computing is constantly on the look-out for original and well-written articles and programs for publication. Feel free to submit your work to us for consideration for publication.

All submitted material must be in machine readable form. This applies both to programs (in any language) and to documentation, which should be prepared with a BBC or Electron wordprocessor. 5¼ inch disc (40/80) or cassette equally acceptable. Please also include hardcopy and any suitable illustration, photographs and/or screen dumps.

If you are considering submitting material to A&B then please send a S.A.E. for a comprehensive style sheet. It's also sensible to give us a ring before going ahead with any major work.

All submissions will be acknowledged and material returned if not required. On acceptance the copyright in such works which will pass to Argus Specialist Publications Limited will be paid for at competitive rates. All work for consideration should be sent to the Editor at the above address.

FEATURES



Twelve December 1985

Published by Argus Specialist Publications Ltd., Number One, Golden Square, London W1R 3AB. Tel: 01 437 0626.

All work for consideration should be sent to the Editor of A&B Computing at our Golden Square address.

REVIEWS

SOFTWARE

A&B Computing is published monthly on the first Friday of the month preceding cover date. Distributed by SM Distribution Ltd., 16-18 Trinity Gardens, London SW9 8DX. Telephone: 01 274 8611. Printed in the UK by Garnett Print, Rotherham and London.

Printed in the UK by Garnett Print, Rotherham and London.

Doubling Up

Tony Self

Simplified volume handling from UDM plus part 1 of the official upgrade from Acorn

UDM UPDATE

In the October issue of A&B,Clive Grace commented in Feedback that the UDM DDFS was one of the best around, but couldn't they do something to clear up some of the gremlins which existed in the version 2.00. Well Clive must be psychic, as UDM have recently released a version 3.1 which should put matters right.

UDM have made some very nice improvements to this version, which certainly makes their system one of the easiest to use for a novice.

All the major changes centre around the expanded volume facility in double density mode — unfortunately, they have done nothing to improve on the maximum number of files available in single density, which still remains at the standard 31.

Version 2.00 gave you the facility to *EXPAND the disc catalogue in double density mode, increasing the maximum number of files to 120, by adding three extra volumes. Each volume was capable of holding 30 files. These could be selected by *VOLUME ‹drv›‹vol› (where vol was a letter in the range A — D). Version 3.1 uses the same system, but makes the volumes easier to access. In fact the user need not know how the volumes are organised at all.

The *EXPAND command has been removed and replaced by an option available after formatting. The formatting command itself has has been improved upon, as it now displays the track being formatted during the operation. I think it is always reassuring to know a program is actually doing something. After the formatting has been completed you are asked whether you require expanded catalogues and then whether you wish to format another disc — again a useful addition.

TRANSPARENT VOLUMES

The best improvements, however, are to the way the system manages the volumes for you. In version 2.00, if you knew a file was on a disc, but were unsure about which volume, you could invoke the *FIND command to search all the volumes for you. On finding the file, that volume would be selected in preparation for a load command. Well version 3.1 does not have this command, instead you just load the file. If the file can not be found on the current volume it automatically selects the next volume and tries again. It does this for all four volumes, before issuing a file not found error. Similarly, when saving files, if the current catalogue is full, it will select the next volume and so on until it finds free space or issues a catalogue full error. In both situations the volume actually loaded from or saved to becomes the current volume. Of course you still have the option of manually selecting the volume with *VOLUME, which now accepts lower case as well as upper, but I suspect that this command will become more or less redundant to most users.

Finally a new command has been added — *CATALL, which, as you might imagine, will list all the catalogues on an expanded disc. This command works in a very neat way. It starts by cataloguing the current volume. Pressing the space bar will then display the next volume's catalogue and so on. However, pressing any other key will exit the routine and leave you in the last volume displayed. The key pressed is also put into the keyboard buffer, so it can form part of the next command you are going to enter (i.e. "C" if you are going to CHAIN a program).

UDM have obviously gone to some trouble to listen to their customers and the comments made in the press, and have come up with a very worthwhile upgraded DDFS Rom. The package now also includes a disc sector editor which operates in either density, an essential tool for the serious disc user.

I would certainly recommend that all existing UDM users upgrade their system and that readers contemplating purchasing disc systems should seriously consider this option.

ENTER ACORN

A little late in the day, but nevertheless welcome, Acorn have brought out their official 1770 upgrade. Although this kit does not provide for double density operation in itself, it is the first step to take if you wish to avail yourself of the Acorn ADFS. Basically the kit replaces the old 8271 and with a disc controller board holding a WD1770 as used in the BBC B+ and many of the independant double density boards around, such as Opus and Solidisk. The kit is very similar as well, comprising the board of two TTL chips, a couple of links (if you are lucky) and a 16K EPROM.

If you read my review in the May issue, you might remember my comments about Solidisk's ingenuity in using shorted polyester capacitors as links. Well Acorn have to take the biscuit. I quote from their covering letter which came with the kit which only included one wire link — "these are in short supply, please make another up using a staple of other suitable material".

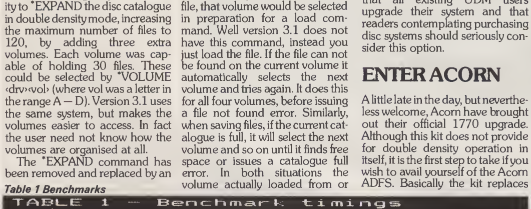
Fitting the kit is very straightforward, helped emensely by the excellent fitting instruction booklet. The DFS EPROM supplied was the version 2.20, which is an update on the EPROM originally supplied with the B+. The documentation supplied with the kit is excellent. The 93 page DFS user guide is the one which has been issued for sometime now, but addendum sheets are included covering the new commands available on the 1770 system.

There is little point in me covering the facilities of this DFS as they have already been covered in articles on the B+. Suffice it to say that buying this upgrade on its own would be pointless if you already have an Acorn DFS, as you will undoubtedly find that some of your existing protected software will not run on the 1770 system. However for new users and those who wish to take advantage of Acorn's ADFS this is probably the right way to go. Also new software becoming available on disc should be written taking into account Acorn's new standard.

I hope that in the next issue I will be able to report on the ADFS itself. Although I have already recieved a copy of the ROM I am still awaiting a copy of the utilities disc and the user guide.

As a final note I have included in Table 1 benchmarks covering these products, plus the figures for the Kenda DMFS, kindly supplied by Allen Hardy, and some mystery figures. Make sure you get next month's issue to see how these figures are obtained.

Table 1 Benchmarks



```
TABLE 1 - Benchmark timings

Benchmark       Acorn     UDM DDFS  3.10     KENDA DMFS 1.04     ????
                single   single    double   single    double

    1            5.59     4.42      4.13     7.17      6.01      0.19
    2            5.39     3.79      3.92     5.56      4.34      0.17
    3            1.87     0.88      0.81    60.58     60.59      0.49
    4            6.95     5.77      5.47    10.96      8.89      4.59
    5            6.63     5.47      5.34     5.78      5.52      2.34
    6           10.57     9.43      8.76    14.82     12.02      6.15
    7            9.97     8.65      8.36     8.93      8.54      2.85
    8            3.58     2.79      2.47     5.41      4.71      1.59
    9            3.23     2.26      2.33     2.60      2.68      1.05
```

View From The Top

Mike Kent

Micros in our primary schools. Are we making the most of them?

Four years ago, throughout England and Wales, a survey of computing in primary schools was carried out. Thirty two schools owned a computer. By January this year the figure had risen to 26,000 machines, a quite staggering increase by any standards.

This doesn't mean to say, of course, that all the computers are actually being used. How many, I wonder, have ended up in stock cupboards after a few teachers have given the proliferation of connecting wires a cursory glance and decided that computing wasn't for them...and thus, the children they teach? Some, I suspect, may still be lying dormant in their cardboard boxes...

Like many thousands of other schools in the early eighties who had decided to take advantage of the Department of Education scheme for purchasing a computer, we approached the idea with a mixture of excitement and apprehension. As a staff, our combined experience and knowledge of computing was minimal, and reaction to the idea of even buying a computer was decidedly mixed, with several teachers suggesting that the money could more profitably be spent in other ways. However, after much discussion, it was decided that our children should indeed become computerate, and that we should learn alongside them.

The first problem came in deciding which machine to buy. The choice under the DoE scheme narrowed it down to three; the Spectrum, the RML 480z, or the BBC. Since the BBC had received outstanding reviews and had been selected by nearly all the authorities as the ideal machine for Primary use, we tended to favour it. However, on learning that ILEA had put its weight behind the 480z, it seemed sensible to take the advice offered, and opt for this machine instead.

The afternoon our Link arrived was an exciting one, as we unpacked the boxes watched by children who no doubt wondered if their teachers were capable of handling this new technology. No technofear for them, of course. Most of them were already cracking up enormous scores with arcade games on their home Spectrums.

Since the hefty instruction manual made as much sense as a quick lesson in brain surgery to us raw beginners, it was some time before we were in a position to actually connect up, let alone compute, and by five o'clock the less faithful had shrewdly departed for home. Undaunted, we put a program into the cassette recorder (discs weren't available to us then), gave the leads a final check...and were astonished to find that it all worked!

Although computers are supposed to be reliant on human input, ours quickly proved it had a mind of its own. It soon refused to load programs reliably (usually just at the last byte!), and loading a program of any length seemed to take forever. It also produced strange messages on the monitor, and in general gave the staff the impression of being as user friendly as a stick of dynamite! The starter pack of software supplied with the machine was another cause for concern. One of the programs was excellent, two were interesting, and the remainder were unbelievably banal... mere fourth rate text book material given the gloss of modern technology.

DOUBTS

Even those of us who were keen to promote computers began to have serious doubts. In order to use the computer, a teacher would have to wheel the trolleyful of equipment to the classroom, set it up, choose a program from the narrow range of passable material and then load it, assuming, of course, that the equipment wasn't having one of its temperamental days. And what of the classrooms not on the same floor as the computer? How would we get the trolley up or down the stairs? How many children would miss out on computing in the course of a week because we only had one machine to share between them all? (I *know* you want a go on the computer, Simon, but we only have *one* for all two hundred of you. Never mind, you'll definitely have three minutes on April 3rd 2010...

For half a term we struggled with our one computer, though it tended to be based in the library and was used mostly by small groups of Juniors. Certainly, some worthwhile work was done, mainly by fourth years with LOGO. Though interest amongst the staff was sustained it was obvious that most children hardly saw the

CONTINUED OVER

machine at all, mainly due to its weight and immobility.

We could have left it at that, but the interest aroused had given us a determination to succeed, and after more discussion we decided to adopt a different strategy altogether, though it meant virtually starting all over again. Since the most successful *home* computer was the Spectrum, why shouldn't we capitalise on that, and use the computer most children would be using at home? The Spectrum was small, inexpensive, very light to move about, and had an excellent specification. It had been criticised, in terms of use by children, for its multi-function keys which might prove confusing, and the small rubber keys which could be awkward to use. But then, we reasoned, we often under-estimate childrens' ability anyway, and it was likely that our children could actually handle multi-function keys better than many adults. The small keyboard problem was rapidy discounted. Children tend to have small fingers!

Since the cost of a Spectrum was relatively small, we then had an exciting idea. Why not equip *every* classroom with a Spectrum, from Reception upwards? Ambitious, certainly, but if we could trade back our expensive LINK 480Z.......

We approached the Computing Inspectorate rather bashfully to put forward our idea, but it was rejected with a firmness that surprised us. The Spectrum, we were told, was an unreliable machine. It woud break down frequently or overheat, it wouldn't stand up to a heavy workload, and it wasn't suitable for "serious computing". Since reports from friends and colleagues who owned Spectrums differed from this view (including our Chair of Governors, himself a keen Spectrum enthuiast) I became convinced that we were on the right track. My philosophy was essentially a simple one; by giving a computer to each class, the children would not only have much hands-on experience, but also see the computer as an extremely useful and versatile tool in their learning.

We began to see the acquisition of our computers as a goal to be aimed at over the period of one year. Each class would need a TV/monitor, a cassette recorder or microdrive for loading, somewhere safe to lock the equipment away at night, and a method of keeping it plugged together on a semi-permanent basis. No teacher would delight in spending an hour before school sorting out a mass of tangled wires before plugging in! We would also need a good basic software library organised on a class-loan basis.

The theory seemed fine, and now we had to fund the project. Every penny was important. Tins rattled conspicuously at concerts, raffles were abundant, and our parents were marvellous, helping us to raise over six hundred pounds on a sponsored walk alone. The total crept upwards, and together with a generous cheque from a friend of the school, we were in a position to equip all our classrooms after just seven months. I disappeared into our craft room for a day with several children, and together we constructed eight wooden trays to hold the sets of equipment. The trays were divided into small compartments to hold each item firm;y in place, and channels were built to contain the wiring. A 4-way 13 amp block was then screwed to each tray.

As the kits arrived, each tray was fitted with a Spectrum, transformer, thermal printer (given free by the distributor), cassette recorder, and demonstration tape, together with the necessary linking leads.

The beauty of this system was that the tray and its contents were very light to carry, even for a child. The equipment, apart from the TV, stayed permanently connected together, and the time taken by the teacher to 'set up' was absolutely minimal. She had only two things to do; connect the mains lead from the 4-way socket to the wall, and plug the Spectrum's aerial lead into the classroom TV, which, of course, could also be used for watching educational broadcasts.

Security for all this equipment might have presented a major headache, but the help and advice given by the security officer soon ensured that each floor of the school had a large safe. A teacher now had only a few yards to walk in order to store her tray and TV at hometime. This,and the ease of setting up, has made certain the equipment is used regularly.

The software library began to expand quickly. Much of the early commercial material for the Spectrum was as dismal as our initial starter pack, but our experience and knowledge were growing..and so was our ability to select a suitable range of programs across the curriculum for both Infants and Juniors. Currently, software has improved significantly in quality, and the best tapes come complete with a workpack of ideas that can occupy children for some weeks.

There were, of course, some teething problems with our equipment, partly due to the very heavy use the computers were receiving from a multitude of tiny fingers! During the first year, three of our Spectrums had to be returned, though they were back within three weeks. We couldn't grumble at that!

So far, then, the children were using commercial software for exploring adventure games, stimulating writing, creating pleasing patterns with a strong maths content via Logo Challenge, learning about a variety of subjects from road safety to ballooning via simulations..and for practising basic skills. There is still much contention about isolating skills acquisition from the modern 'broad based' primary curriculm, but it seems logical to assume that children can hardly have an autonomous approach unless they have a healthy 'tool kit' of basic skills with which to explore their environment, and the better 'skills' software tapes can make this a very enjoyable experience.

Naturally, care needs to be taken in selecting software. I recently saw a program which gave the child a couple of minutes to do his tens and units before plunging the world into nuclear war.....

NEXT STAGE, FIRST STEP

Since the revised computer project had taken off so well, in a shorter time than expected, we decided to move a step further. A short while before our project had been initiated, I had bought a BBC Micro for home use, and a colleague had bought a Spectrum. We tended to arrive for school rather tired in the mornings, having spent long nights delving into the fascination of BASIC and making the first hesitant steps towards writing our own educational software. It felt rather like learning a foreign language; tortuous at times but worth it when you got there! From discussing the relative merits of Spectrum versus Beeb, it became obvious that the BBC machine had a lot to offer. It was expensive, and prohibitively so in any quantity, but since the bulk of the quality educational markey was aimed at the BBC, we wished to take advantage of it.

During the course of the next year, we spent part of our school allowance on three BBC's, and after fitting them with Wordwise chips, we were in a position to introduce the children to word processing.

Wordwise is a thoroughly comprehensive word processor, but simplicity itself for children to use as well. After learning a handful of commands, the child can type out his writing, edit it, correct spellings quickly, and experiment with moving sequences of text around. With poetry writing, for example, the word processor really comes into its own. Anyone who has watched a child's face as his or her piece of writing emerges from a printer for the very first time will understand the importance of word processing and the skills it encourages. It is not, of course, a substitute for calligraphy, but an important and exciting extension of it. Since the BBC has such a user-friendly and robust keyboard, children take to typing quickly and easily on it.

It is important to remember that the staff were experimenting and learning, alongside the children. Though my own fascination with computing had begun to fill many of my leisure hours, a project of this kind has to be nurtured slowly. Many teachers are still wary of anything more technical than a tape recorder, though at our own school help is always at hand, and nobody talks in machine code rhetoric. We feel computing must be seen as an activity for all, rather than a technical elite who are able to jargonise skilfully but say very little.

What, then, of the future, for we mustn't simply be content with putting computers within easy reach of our children. There are, I feel, many exciting avenues to be explored.

At present, a group of Juniors spend a morning with me, learning how to write their own programs. BBC BASIC is a very structured dialect, and pleasantly logical to use, but even so I was surprised at how quickly the children began to absorb it. They have experimented with colour and sound, graph work and number crunching, design and animation, and written lots of little routines just for fun...like finding out how much pocket money they are likely to earn over the next decade! It won't be long before they are able to write useful little learning programs for the Infants to try. Certainly, the logic and reasoning power developed by program writing is powerful and readily apparent, stimulating a whole range of new skills.

We must move steadily into the realm of data basing, with its opportunities for classifying, sorting and comparing. We must bring the children to a further understanding of how influential computers are on their lives, and how useful a tool they can be. We must investigate the growing range of peripherals for the computer, which can be helpful to both teacher and child, and we must look to the time when children will not only build models of towns, traffic lights, bridges and lighthouses, but hook a computer up to them and make them work.

We can even extend computer learning into the Nursery, too. At the age of three, my youngest learned most of her alphabet, and how to spell her name, by playing on the computer.

As we have discovered, it is important to make the computer an item of everyday classroom equipment. This, however, is merely the first step.

Mike Kent, Headmaster of Comber Grove Primary School in London, argues that regular hands-on experience is essential for young children, and describes how this was achieved in his own school.

Down to Business

Jon Vogler

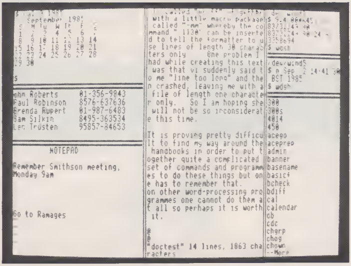
UNIX is the most powerful operating system ever to run on a micro. *Torch* offer it on the ultimate BBC add-on — but not for the faint-hearted.

INTRODUCTION

"I shan't get a car; it's far too difficult to learn to drive." Ever heard that from a businessperson? Under the age of sixty I mean? Perhaps from the bicycle wobblers you pass on your way to work. For most people however, the first few nervous lessons behind the driving wheel, the complexities of gears and clutch, handbrake and hill starts and the tedium of the highway code are worth learning because of the increased speed and capacity and power that a car provides. Trouble is, even cars are not all that fast, especially in traffic jams and by the end of a long trip you are fit for nothing.

You can go by train, relax, stop thinking too hard. There is just as much speed, capacity and power but something else is lost: flexibility. The rails are fixed and someone else plans the timetable without considering your particular needs. All too often, this means you cannot get to exactly where you wish at the time you want, even if you get up at four o' clock in the morning. What is the answer then? Some tycoons are so impressed at the improvements which technology can make to their personal convenience and efficiency that they buy and learn to pilot a personal airplane or helicopter. These people arrive for meetings ontime, fresh, confident. We are no longer surprised that business people should invest thousands of pounds and hundreds of man hours learning time to achieve one ultimate goal: high personal productivity. Why then do these same sultans of business say "I shan't get a *UNIX* system; it's far too difficult to learn"?

Of course many business folk are still at the bicycle stage: their offices use typewriters, filing cabinets, even slide rules or adding machines and those telephones that wear a groove round your index finger. Many who decide they needed the power and flexibility of the micro computer (the office equivalent to the motor car), use *CP/M 80* or *MS-DOS* operating systems and a generation of business programs such as *Wordstar*, *Visicalc*, *Dbase II* and *Perfect Software I*.



```
$ cal 9 1985
   September 1985
 S  M Tu  W Th  F  S
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30

$
```

```
John Roberts      01-356-9843
Paul Robinson     8576-637636
Brenda Rupert     01-987-6483
Sam Silkin        8495-363534
Len Trusten       95857-84653
```

```
        NOTEPAD

Remember Smithson meeting,
Monday 9am



Go to Ramages
```

```
 with a little macro package
 called "-mm" whereby the co
mmand ".1130" can be inserte
d to tell the formatter to u
se lines of length 30 charac
ters only     One problem I
had while creating this text
 was that vi suddenly said t
o me "line too long" and the
n crashed, leaving me with a
 file of length one characte
r only.   So I am hoping she
 will not be so inconsiderat
e this time.

It is proving pretty difficu
lt to find my way around the
 handbooks in order to put t
ogether quite a complicated
set of commands and programm
es to do these things but on
e has to remember that.
on other word-processing pro
grammes one cannot do them a
t all so perhaps it is worth
 it.

@
@
"doctest" 14 lines, 1863 cha
racters
```

```
$ 9.4 00**45
83731143 00
83731124 00 24
3358694
$ wdsh
```

```
/dev/wind5
$ n Sep  2 14:41 30
 BST 1985
$ wdsh
```

```
300
300s
4014
450
acego
aceprep
admin
banner
basename
basicf
bcheck
bdiff
cal
calendar
cb
cdc
chgrp
chog
chown
--More
```

1. The Unicorn's window facility: seven live tasks all running concurrently and more can run in the background.

They are powerful and flexible but have one problem, you must learn to drive them just as you had to learn to drive a car. Those who persevere find it rewarding, but many funk the effort. They want more speed and power but are unwilling to do the learning and this has provoked the birth of very user friendly operating systems such as *Gem* or *Lisa* which soak up spare memory to create pictures and a clickety mouse to drag them around the screen. Like the train, this approach has one huge disadvantage: although you still get the power and speed and can run everyone's favourite, business program (you know, the ones with trendy names like *Symphony*, *Framework* or *Jazz*), what is missing is flexibility. If what you want to do is that little bit different; if your business doesn't quite fit the pattern or if you are more adventurous and want to go where others have laid no rails, you find yourself at a dead end.

The other alternative is *UNIX*, which is the equivalent of the personal helicopter. *UNIX* is not a program, or a computer; it is an "operating system", just like *CP/M* or *Gem* or the BBC's own. The difference is that *UNIX* is the most powerful one ever to run on a micro. Not only will it do anything that you wish and go anywhere that you like but taking your employees or business partners along for the ride costs little extra. There is almost total flexibility and huge amounts of power and capacity but both the helicopter's disadvantages: it is expensive and learning to use it needs confidence and a lot of time.

To see whether it was worth the money, and whether its difficulties and complexity had been exaggerated, I wanted to try it myself. Impossible for the owner of a modest 32K *Beeb*? Not at all, because *Torch* offer the ultimate in BBC add-ons: the *Unicorn* which hangs a *Motorola 68000* 32-bit chip (don't worry about the numbers that just means "very powerful indeed!") and a megabyte of memory (that just means 30 times as much as the normal *Beeb*!) and a 20 mega byte hard disc unit (40 times as much as an average floppy disc drive). The advertisements were most persuasive: a system that would run half a dozen different operations at the same time; would display them in separate windows on the screen; could be networked to ten other people who need each have no more than a *BBC B* with an *Econet* fitted and a monitor. All this at around the price of a (20 megabyte) hard disc *ACT Apricot XI* with half as much random memory and no real multitasking capability.

My especial interest was that *UNIX* seemed to solve so many of my current problems. How to look up someone's telephone number on my database when in the middle of word processing this article How to extract information from the database and put it directly in a report without tedious and borate programming beforehan How to look up my floppy disc alogue or my bibliography or the calendar for 1988 while compili a budget spreadsheet? How insert a couple of tables of data the text of a technical report? F tastic if these can be achieved what about the penalty? w about that notorious unfriendliness? *UNIX* is huge the Torch version comes with 1000 different files and prog (not bad value, around £4 inclusive of hardware) and hand books are about the size the London telephone direct So readers will forgive me if article is split. This month I s describe the standard *UNIX* tem and next month discuss it in business and running applications packages on it.

SOME ENCOURAGEMENT

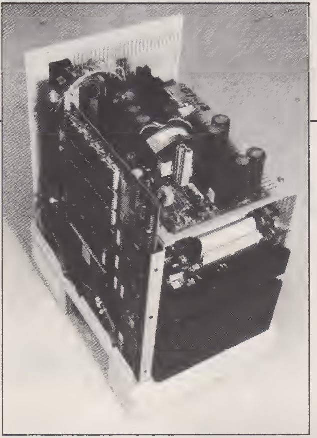
In case it proves difficult describe this highly technical tem , let us start with some ragement. Fig 1 shows what m tasking windows can really offer a business person. The central window contains the text of an article for *A&B*.

To the left, starting from the top, this month's calendar, so I do not miss your editor's merciless deadline. As well as this year it seems to cope very amply with the year 2010. Then a window on the database giving one or two phone numbers (I needed them in a hurry for someone who just telephoned). At the bottom, my "jotting pad" notes made during the phone call, for later reference.

On the right of the screen: at the top the calculator, on which I could work out my income tax or how long it will take the probe to reach Haley's Comet. The calculator program seems quite comfortable working to twenty six decimal places!

Below is today's date and time and, at the bottom a list of disc files, so I can check the name of a file I wish to insert in the article. These are not just dead information on the screen: each is a live program or process currently running.

As well as the seven windows I could have other tasks grinding away in the background: for example calculating a complicated formula that needs to reiterate many times or formatting a document for printing or counting the number of words in an article. You don't have to stop and wait until these processes have finished. Your printer can be printing away in the background and another program could be transmitting data over the telephone or interrogating a data base in California. Think how long it would take, on a normal BBC, to gather all this information and perform all these tasks. That's the high personal productivity that *UNIX* offers.



2. Inside the cabinet of the Torch Unicorn.

SOME FEATURES OF UNIX

I shall not try and describe how *UNIX* works. That needs a whole book and some people get quite nutty about it. *UNIX* is full of expressions like the "shell" and the "kernel" which we don't need to go into. Rather let's describe the hardware you get if you buy *Torch UNIX*. You need the 20 megabyte fixed disc (reviewed in *A&B Computing July 1985*); *UNIX* will gobble between half and two thirds of it. That still leaves plenty of capacity for your *CPN* files and programs (CPN is the CP/M look-alike operating system under which the Torch Z80 runs Wordstar, dBase II, Perfect Writer or whatever); you don't lose access to these because you have *UNIX*.

Contained inside the hard disc cabinet is the Atlas printed circuit board which contains both the 68000 chip and also the Z80 chip (Fig 2). Also within the same box is the huge random access memory: you have the options of half a megabyte (sufficient for a single user) or a full megabyte (Fig 3), necessary to rún an efficient multi-user system. One of the reasons why *UNIX* can run so many processes simultaneously is that all this vast brain is used for the actual processes that are running. *UNIX* loads them from the disc and dumps them again when they are finished so you don't have memory cluttered up with processes which are not actually running. Finally you need a *UNIX* ROM to be fitted in your *BEEB* and, if you want a multi user system, an *ECONET* network. The various costs are shown in the inset box Fig 4.

It is the terminals which are so remarkably economical. They don't even need a second processor: a straight *Beeb* fitted with Econet and a special ROM is quite sufficient. Once *UNIX* is running on the host machine, any terminal can link in quickly and easily. Nor incidentally do you lose access to those beloved BBC games programs. Keys *B* and *Break* will still get you back to the BBC operating system. However, although you can multitask (run several tasks simultaneously) from the terminal, under the Torch system the terminal will not work with separate windows; this facility is only available on the host machine. However, this might even be a good thing, you don't want your secretary writing out his (or her!) weekend shopping list while typing your monthly production report nor the warehouse clerk making changes in the sales ledger. In fact there are no problems of this kind, because *UNIX* offers a sophisticated protection system. Each file can have three different sorts of protection, read, write or execute for three different groups of people: the owner, a defined group to which the owner belongs or anybody else (Fig 5).

UNIX AND THE USER

UNIX is most often criticized for being unfriendly. People miss the neat displays and helpful error messages so familiar from the BBC. Partly this is because *UNIX*, from within the system, does many of the things for which, with just a BBC, you would need to buy a special program. There is nothing that prevents extravagant displays (except prudent economies of disc and memory space, sometimes ignored in the face of such abundance!) However the lack of error messages is definite. *UNIX* is very terse. If it can't find a file, it doesn't say *"FILE NOT FOUND"* ,it just returns to the prompt — rather like those travelling companions abroad who, as you pass some majestic scene or fascinating old building merely draw on their pipes and grunt!

The unfriendly feeling is emphasized because *UNIX* does every thing in lower case letters and uses extremely concise commands. However this too has merit: it saves one finger constantly hovering over the shift key and reduces the number of key strokes. It is only slightly quicker to run a game by typing in small letters "aliens" than by the familiar BBC: *CHAIN "B:ALIENS"* but when such commands are repeated hundreds of times the difference becomes significant. In fact many of the commands are helpfully mnemonic. If you want to count the number of words in a file called "fred", you simply type in *wc fred*. Likewise the massive handbooks actually look less friendly than they are. They are looseleaf, extremely well subdivided, comprising both separate sheets on each command plus explanatory articles (mostly written by the men who invented *UNIX* at the Bell laboratories in America in the late seventies) and although some are highly technical, there are others designed for the nontechnical and the new user.

Just like using a motor car service handbook, you don't need to understand all the technicalities of the engine to find the bit that tells you how to top up the battery. But

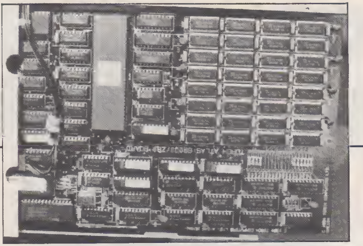
CONTINUED OVER

if you are a motor mechanic, then you want all the engine detail you can get. However many users will find, as I did, it is helpful to buy one of the many "Teach yourself" books. *Torch* list eight of them and my local bookseller offered a further five!

THE UNIX FILE SYSTEM

Another feature that many people find difficult at first is the concept of "path names". This is also used (but is less essential) by computers (such as the *IBM PC*) that use MS-DOS. Instead of dividing the hard disc into a number of surfaces (B:, C:, etc just like floppy discs) *UNIX* treats the whole huge area as one. That brings the immediate, enormous advantage that you don't have to remember (or care) on which surface your program or file is located. Hundreds of files are all equally accessible. But, as always, there is a penalty to be paid. How do you locate the exact one of a thousand files you need? Imagine you have just arrived in Britain and want to find someone called *Jon Vogler*. You could go to some central government office and look up all the records of all the people who have ever been born in Britain and you might, with difficulty, track him down. Much easier however if you have been given an address. In fact we address letters the wrong way up: to find *Jon Vogler*, the post-office starts at the bottom and progresses upwards. The first information, that he lives in *West Yorkshire*, narrows the search; *Leeds*, as the town in *West Yorkshire*, narrows it further and so on. In fact we could write his address more logically as **West Yorkshire/ Leeds/ Roundhay/ The Avenue/ 40/ Vogler/ Jon.** If there was another Jon: *Jon Smith*, living at *18, The Beeches, Ruislip*, he would be clearly distinguished as **Middlesex/ Ruislip/ The Beeches/ 18/ Smith/ Jon.**

Unix path names work exactly the same way. The place you start from (*Britain* in our example) is known as *root* — like the root of a tree. Then you follow a path name through various branches to get to the file you need. Each of the files can be regarded as a fruit hanging on the tree and various limbs, branches and twigs, from which the fruit hangs, are called "directories". One directory may have several files hanging from it or may have some files and also some sub-directories. A typically *UNIX* path name is **/usr/bin/wc.** The first slash indicates the root. *usr* is like a branch and *bin* a twig (they stand for user and binary and are directories that are present in the *UNIX* structure when you buy it. *wc* is the fruit hanging at the end of the twig, the wordcount program. You can also create your own directories and files.



3. The Unicorn's main board: the thirty two memory chips (top left) give a megabyte of RAM; the huge Motorola 68000 chip (lower centre left) has 64 pins to handle 32 bit words.

Supposing in your business the *UNIX* system is going to be used by Tom and Betty. You might create a new directory, called *new*, which would contain all the files added since *UNIX* was received from the supplier. Then in *new* you might create three more directories: one called *common* that anybody could use, one called *tom* and one called *betty* for their sole use. Now suppose both Tom and Betty create a file called *data*, how does the computer know which one to use? Why, because one has the pathname */new/tom/data* and the other has the pathname */new/betty/data*. That is all there is to a pathname, it is simply a file with a rather complete address which enables the computer or its user to find any program or file without difficulty. And of course Betty could lock her data file in such a way that perhaps either she or Tom could both read it but only she could write to it to make alterations.

Every *UNIX* system has also one "super user" who can read or alter every file. And of course *UNIX* offers the normal range of commands to move files from one directory to another, copy them, rename them and alter them in various ways and has a powerful series of "wildcards" so that, for example, specifying the pathname */new/betty/** can refer to every file in the directory */new/betty* if you wish to list or protect them all.

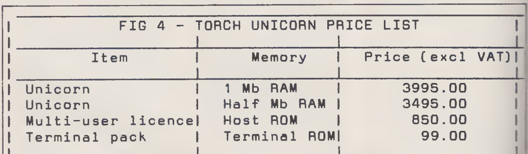


FIG 4 - TORCH UNICORN PRICE LIST

Item	Memory	Price (excl VAT)
Unicorn	1 Mb RAM	3995.00
Unicorn	Half Mb RAM	3495.00
Multi-user licence	Host ROM	850.00
Terminal pack	Terminal ROM	99.00

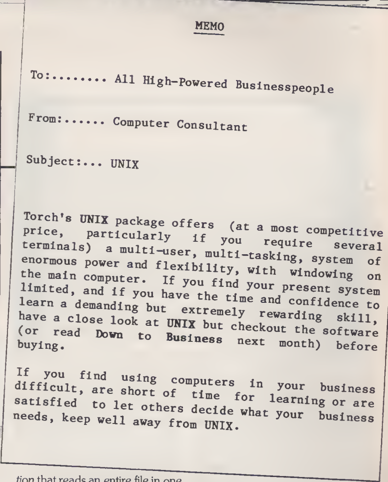
4. Prices of the Unicorn

PROCESSING PROGRAMS AND FILES

One reason why *UNIX* is so flexible is that you can do so many things with programs and files. I have already mentioned running them in the background while you do other tasks on the screen. You can also *redirect* them. Suppose you have a file containing a list of names of employees, arranged in order of the date they joined the company and you want to sort it into alphabetical order. The *input* is the list of names, the *process* is to sort but what about the *output*: do you just want it to appear on the screen? Not much you can do with it there. *UNIX* gives you complete freedom to redirect it as you wish: either to the screen or to a printer or to another file or down the telephone wires to a computer in your head office a hundred miles away.

Have you ever tried to make an alphabetic listing of all your BBC disc files? If so you will know how frustrating it is . You can display a catalogue of one disc on the screen and you can print it but you cannot merge it with the contents of twenty other discs or sort them alphabetically without buying a special program (a program which incidentally I have yet to see produced; anyone got one?) *UNIX* makes this easy, without any special program. I was able to type in one single line: **find / -print>/user/jon/filelist** This listed all the 1250 files and directories on the system by name and wrote them into a file (called *filelist* in my personal directory */user/jon*. Alternatively I could have routed them to the printer or put a *tee* (*UNIX* is full of plumbing words the system so they went both to the printer and to the file at the same time.

However, suppose I did not want to list them but only to count them. *UNIX* will *pipe* processes the output from one process flows into another. The command pipe is the simple vertical bar and I could have piped the output from the file listing described



MEMO

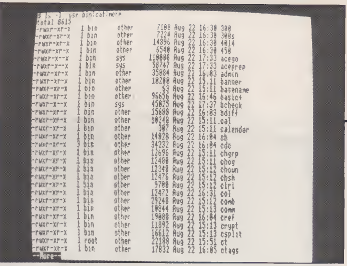
To:........ All High-Powered Businesspeople

From:...... Computer Consultant

Subject:... UNIX

Torch's **UNIX** package offers (at a most competitive price, particularly if you require several terminals) a multi-user, multi-tasking, system of enormous power and flexibility, with windowing on the main computer. If you find your present system limited, and if you have the time and confidence to learn a demanding but extremely rewarding skill, have a close look at **UNIX** but checkout the software (or read **Down to Business** next month) before buying.

If you find using computers in your business difficult, are short of time for learning or are satisfied to let others decide what your business needs, keep well away from UNIX.



```
$ ls -l usr bin:cat:more
total 8615
-rwxr-xr-x   1 bin     other      7108 Aug 22 16:30 300
-rwxr-xr-x   1 bin     other      7224 Aug 22 16:30 300s
-rwxr-xr-x   1 bin     other     14896 Aug 22 16:30 4014
-rwxr-xr-x   1 bin     other      6540 Aug 22 16:30 450
-rwxr-x--x   1 bin     sys      110008 Aug 22 17:33 acego
-rwxr-x--x   1 bin     sys       58747 Aug 22 17:33 aceprep
-rwxr-xr-x   1 bin     other     35084 Aug 22 16.03 admin
-rwxr-xr-x   1 bin     other     10200 Aug 22 15.11 banner
-rwxr-xr-x   1 bin     other        63 Aug 22 15:11 basename
-rwxr-xr-x   1 bin     other     96656 Aug 22 16:46 basic+
-rwxr-x--x   1 bin     sys       45025 Aug 22 17:37 bcheck
-rwxr-xr-x   1 bin     other     15688 Aug 22 16:03 bdiff
-rwxr-xr-x   1 bin     other     10248 Aug 22 15:11.cal
-rwxr-xr-x   1 bin     other       307 Aug 22 15:11 calendar
-rwxr-xr-x   1 bin     other     14028 Aug 22 16:04 cb
-rwxr-xr-x   3 bin     other     34232 Aug 22 16:04 cdc
-rwxr-xr-x   1 bin     other     12696 Aug 22 15:11 chgrp
-rwxr-xr-x   1 bin     other     12480 Aug 22 15:11 chog
-rwxr-xr-x   1 bin     other     12340 Aug 22 15:12 chown
-rwxr-xr-x   1 bin     other     12476 Aug 22 15:12 chsh
-rwxr-xr-x   1 bin     other      9700 Aug 22 15:12 clri
-rwxr-xr-x   1 bin     other     12472 Aug 22 16:31 col
-rwxr-xr-x   1 bin     other     29240 Aug 22 15:12 comb
-rwxr-xr-x   1 bin     other     10844 Aug 22 15:13 comm
-rwxr-xr-x   1 bin     other     19000 Aug 22 16:04 cref
-rwxr-xr-x   1 bin     other     11892 Aug 22 15:13 crypt
-rwxr-xr-x   1 bin     other     16612 Aug 22 15:13 csplit
-rwxr-xr-x   1 root    other     22188 Aug 22 15:51 ct
-rwxr-xr-x   1 bin     other     17832 Aug 22 16:05 ctags
--More--
```

5. File list. The group on the left indicates protection. These files can be read, written or executed by the "owner", executed or read, but not written, by any member of her "group" but only read by anybody else.

above into the wordcount program just by writing: **find / -print/wc.** And if I wanted to sort them into alphabetical order and present them on the VDU, one screenful at a time without "scrolling", I could pipe them through two further processes like this: **find / -print/sort/more.** Now supposing that I had some operation of this type which I wanted to do 10 times a day for a month. Rather than type that rather tedious line in every time, probably getting it wrong now and then, I could turn it into a *shell script*. All this involves is saving it in a file and giving the file a name; in this case let us call it *filecount*. By typing **sh filecount ‹CR›** the *UNIX* shell (which we haven't explained but it is obviously determined to be mentioned!) would run the sequence of commands of its own accord. Suppose however, I only wanted to run the process on working days, not at the weekends. *UNIX* keeps the date permanently recorded (provided you set it correctly when you get started) and, within the shell script system, is a programming language that uses such useful tools (familiar to all who write programs in BASIC) as *if – then loops*, *while*, *for* and *until* commands and the rest. So I could easily include in the shell script a clause that says: Only run *filecount* if the day of the week is Monday – Friday. Imagine the length of BASIC program needed to handle that lot. *UNIX* does it in a couple of terse, economical lines but there are no helpful menus; the user, like any helicopter pilot, has to remember precisely which commands to issue to the machine. That is why, to many, it appears unfriendly.

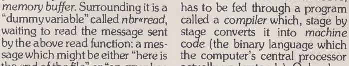
THE C PROGRAMMING LANGUAGE

UNIX is not simply the most powerful operating system available on any micro today. It also includes *C*: one of the most economical and powerful programming languages available. Most modern business applications programs are written in *C* but *C* is also a powerful "number crunching" language. Although *C* can be run on *CP/M* or *MS-DOS* or various other operating systems (but not on the BBC'S operating system) it was invented by the same team who developed *UNIX* and grew up hand-in-hand with it. Most of *UNIX* is written in the *C* language and *C* was developed to operate within the *UNIX* operating system.

C is not a complicated language to learn, particularly if you already know *BASIC* and understand such concepts as loops (*for ...next*) and functions. One of the best books – *Understanding C* by Bruce Hunter, published by *Sybex*, runs to about 300 pages and can be mastered if you spend 2 hours a night on it for a week or two. The reference manual to the *C* language, included within the *UNIX* hand books, extends to only 30 pages and the list of key words is very short.

C programs can be very concise: *C* does in one line what may take a dozen lines of code in *BASIC*. For example a single line of program such as: **while (nbr-read = read (fdi, buffer, BUFSECTS))** is a *while loop* which does several tasks. In the centre is a *read function* that reads an entire file in one pass and puts its contents into a *memory buffer*. Surrounding it is a "dummy variable" called *nbr«read*, waiting to read the message sent by the above read function: a message which might be either "here is the end of the file" or "an error has been encountered". Finally, on the outside, is the *while loop* itself, which keeps the *read function* going and makes it stop when it reaches the end of the file.

C is a compiled language: the program or *code* that you write, in words such as those shown above, has to be fed through a program called a *compiler* which, stage by stage converts it into *machine code* (the binary language which the computer's central processor actually understands). Only when this has been done for the whole program can it be run. (*BASIC*, in contrast, is an *interpreted language*: it is converted to machine code one line at a time). The advantage is that *C* runs very much more rapidly and uses very little

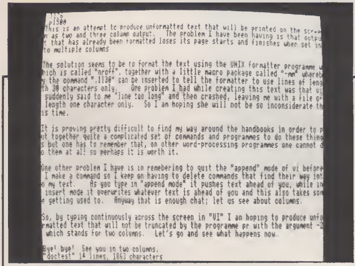


```
INPUT TEXT TO PROOUCE FIG 4 USING tbl

.TS
center doublebox tab (@);
c s s
c | c | c
l | l | n.
FIG 4 - TORCH UNICORN PRICE LIST
_
Item@Memory@Price (excl VAT)
_
Unicorn@1 Mb RAM@3995.00
Unicorn@Half Mb RAM@3495.00
Multi-user licence@Host ROM@850.00
Terminal pack@Terminal ROM@99.00
.TE
```

6. Table pre-processing program "tbl" produced the table in Fig 4 from this input.

CONTINUED OVER



```
 11.6
 p1500
This is an attempt tc produce unformatted text that will be printed on the scre
n as two and three column output.   The problem I have been having is that outpu
t that has already been formatted loses its page starts and finishes wher set in
to multiple columns

The solution seems to be to format the text using the UNIX formatter programme w
hich is called "nroff", together with a little macro package called "-mm" whereb
y the command ".1130" can be inserted to tell the formatter to use lines of leng
th 30 characters only.    One problem I had while creating this text was that vi
 suddenly said to me "line too long" and then crashed, leaving me with a file of
 length one character only.   So I am hoping she will not be so inconsiderate th
is time.

It is proving pretty difficult to find my way around the handbooks in order to p
ut together quite a complicated set of commands and programmes to do these thing
s but one has to remember that, on other word-processing programmes one cannot d
o them at all so perhaps it is worth it.

One other problem I have is in remebering to quit the "append" mode of vi before
 I make a command so I keep on having to delete commands that find their way int
o my text.    As you type in "append mode" it pushes text ahead of you, while in
 insert mode it overwrites whatever text is ahead of you and this also takes som
e getting used to.   Anyway that is enough chat; let us see about columns.

So, by typing continuously across the screen in "VI" I an hoping to produce unfo
rmatted text that will not be truncated by the programme pr with the argument -2
 which stands for two columns.   Let's go and see what happens now

Bye! bye!  See you in two columns.
"doctest" 14 lines, 1863 characters
```

7.The screen editor "vi" uses the entire screen for text.

memory. List it on the screen and the compiled program appears as a confused jumble of codes, so it is far harder to pirate. The disadvantage is that it is tedious to correct the code because, for every alteration, you have to recompile it before testing whether it runs correctly. That, of course, is where the *Torch Unicom's* multitasking windows facility really comes into its own: you can write the code in one window, compile it in the next, and run it in a third, as near simultaneously as you, the user, can manage. For any business that cannot find, or cannot afford, tailor-made software, the ability to write in C is an enormous asset: but only for those with the time and ability to exploit it.

TEXT PROCESSING FACILITIES

Because *UNIX* was originally written to help the engineers at Bell Telephone Labs produce their technical reports, it is particularly strong on text processing facilities. However, these are not like the accustomed wordprocessing packages, such as *Wordstar* or *View*, with all the facilities rolled up behind one single menu.

Instead there are many different components: a variety of editors, two different formatters and several subsidiary programs with special uses. Some of the editors deal with only one line at a time and are virtually obsolescent in these days when we have all become used to full-screen editors. However *UNIX* offers an excellent one called *vi*. To format for a line printer there is a program called *nroff* (pronounced "enroff") and there is a program for formatting photo-typesetting call *troff* ("teeroff"). Any reader who has discovered how difficult it is to get your friendly neighbourhood printer to accept text produced on less powerful word processors should be especially interested in the latter. Both *nroff* and *troff* are difficult programs to use so *UNIX* provides a set of "macros" which tailor them for the average user. In addition there are "pre-processors" for tables of data and for mathematical formulae. Again, anyone who has experienced the problems of incorporating these simply and fluently in word process text will heartily rejoice. The table program is elegant: Fig 6 shows the text I typed in to produce the pricelist table, Fig 4 .

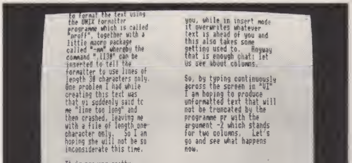
vi

vi has everything you could ask for from a screen editor except one: familiarity. It seems to work quite differently from most programs and it is tempting for the new user to reject it because of this. Typing in is done in a quite separate mode from other activities, although only a single keystroke is needed to switch between the modes. For example if you are in *text mode* and enter the simple command to save your text *":w"* without first touching the escape key to go into *command mode*, then you find you have written ":w" into the middle of your text and, only after some days of practice is it trouble-free to remove it again!

However, once these teething troubles are past, it is a delight. First of all there is a complete range of cursor movements: forwards or backwards by the next letter, word, line, sentence, paragraph, half-screen, screen, section (a section is a portion of text begun with a subheading) and the whole file. You can specify whether your move is to the start or end of the word and can include punctuation or skip over "white space": the net result is that one keystroke serves where other word processors need two or even several and this, as readers will now know, is Vogler's acid test of good editors. One can also move to a specified letter or word or line number. (Lines can be numbered automatically without interfering with the editing). In addition, by prefixing the command with a number, you can move by 5 words, 3 lines, 7 screens etc. There is plenty of screen to move around, because, in contrast to many word processors, (particularly those designed for the IBM PC!) *vi* uses the whole screen depth of 32 lines and the whole width (Fig 7). There is no "wordwrap" but don't worry: *nroff* supplies the necessary justification, filling and hyphenation if required. By prefixing with a "d" any of the movement commands you can delete the text across which the cursor moves. If you want to change a word rather than simply delete it, prefix the movement command with small "c" and you find yourself typing over the unwanted words or sections, marked at the end with a dollar sign to indicate the extent of the deletion, which is completed when you press *escape*, even if the text you are adding is much shorter than that you are taking out.

The arrangements for copying and moving text are particularly powerful because, unlike so many programs which offer only one or at best two or three markers, *vi* gives you 26 markers: an "m" followed by any small letter of the alphabet. To home on a marker you can either prefix the its letter with a "£" sign; or with a single inverted comma sign "'" to return to the start of the line on which it is contained.

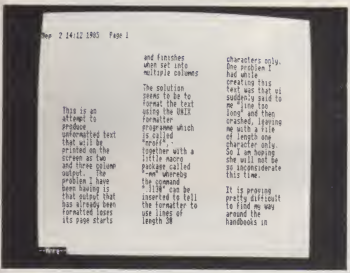
To complement the 26 markers there are 27 buffers: one buffer into which any deletion is saved (without you needing to do anything) and the others which need to be specified by a letter of the alphabet. The joy of the unnamed buffer is that, if you carry out deletions and then regret them, the command small "u" restores you to the condition before you started deleting. Of course if you forget what text is in which buffer you would simply open a small window and list them all to jog your memory. Having deleted or copied text



```
to format the text using            you, while in insert mode
the UNIX formatter                  it overwrites whatever
programme which is called           text is ahead of you and
"nroff", together with a            this also takes some
little macro package                getting used to.   Anyway
called "-mm" whereby the            that is enough chat; let
command ".1130" can be              us see about columns.
inserted to tell the
formatter to use lines of           So, by typing continuously
length 30 characters only.          across the screen in "VI"
One problem I had while             I am hoping to produce
creating this text was              unformatted text that will
that vi suddenly said tc            not be truncated by the
me "line too long" and              programme pr with the
then crashed, leaving me            argument -2 which stands
with a file of length one           for two columns.   Let's
character only.   So I an           go and see what happens
hoping she will not be so           now.
inconsiderate this time.

It is proving pretty                Bye! bye!  See you in two
difficult to find my way            columns.
around the handbooks in
order to put together
quite a complicated set of
commands and programmes to
do these things but one
has to remember that, on
other word-processing
programmes one cannot do
them at all so perhaps it
--More--
```

8.The text from Fig 7 converted to two columns by "nroff", the formatter.



```
Sep  2 14:12 1985   Page 1

                    and finishes        characters only.
                    when set into       One problem I
                    multiple columns    had while
                                        creating this
                    The solution        text was that vi
                    seems to be to      suddenly said to
                    format the text     me "line too
This is an          using the UNIX      long" and then
attempt to          formatter           crashed, leaving
produce             programme which     me with a file
unformatted text    is called           of length one
that will be        "nroff",            character only.
printed on the      together with a     So I am hoping
screen as two       little macro        she will not be
and three colume    package called      so inconsiderate
output.  The        "-mm" whereby       this time.
problem I have      the command
been having is      ".ll30" can be      It is proving
that output that    inserted to tell    pretty difficult
has already been    the formatter to    to find my way
formatted loses     use lines of        around the
its page starts     length 30          handbooks in

--More--
```

9. The same text in three columns.

into a buffer, you can replace it wherever the cursor is and there is also a facility to replace it several times: very useful when writing legal or contractual documents, all of which begin "the contractor shall........"

Most word processors have a facility that enables you to centre the work area (eg the cursor) in the centre of the screen; *vi* goes one better (or two actually) and allows you to postion it either at the centre or at the top or bottom of the screen. Searching and replacing are also easy, with the facility of doing them in reverse and for continuing the search after the first find with a single key stroke. Searches can be instructed to ignore the case (ie treat "happy", "Happy", "HAPPY" as the same). One very useful facility enables you to match braces, ("" "" () " [ ]). When your cursor is on the opening bracket, typing the command "%" will move the cursor to the matching bracket, particularly useful when writing programs.

Working with multiple files also has undreamed of power. If you want to insert part of one text into another, you just display line numbers and specify the range of line numbers to include. Likewise you can save part of a text or edit only part. If you are really chopping and changing texts about, you could run them in two or three windows with all the line numbers showing in order to carry out a rapid mosaic operation.

vi makes it easy to set out text on the screen: a double sideways arrow will move the complete line of text in either direction and there is a setting command to change the "shift width" distance by which it moves and also to auto indent the following lines if required.

However to me the greatest of all joys with *vi* was that, while still remaining in the word processor, one can still do those useful things which are normally denied. You can set the function keys and this enables you to make full use of them for writing repetitive phrases. However, *vi* has an additional feature, of mapping not only the red function keys, but any key on the keyboard, to contain any phrase or command that you may wish. This gives terrific economy with commonly used phrases or commands. Finally while still leaving the text on the screen you can leap out of word processing, perform a calculation, check on a filename, interrogate a data base, or do anything else you like and then leap back into your word-processing text without losing so much as a character.

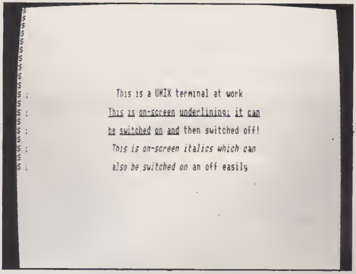
If *vi* is powerful, *nroff* and *troff*, when combined with their various macros, are even more so. They can be summarised in one phrase: anything you can do with words on paper, the *roffs* can do it. The facilities I particularly valued are of adjusting the environments for headings. You can set your own combinations of underlining, bold type, number of lines before or after, indentation etc., and the facility to print text in as many columns as you wish (Figs 8 and 9). It is simple to view the text on the screen before printing: there is none of that nightmare of trying to preview text on the screen, one hand firmly fixed on the shift and control keys, and then being unable to go backwards to see what it said on the previous page. Moreover the underlining and italics can also be viewed on the screen (Fig 10).

One serious deficiency with the text processing facilities is the lack of an adequate manual index. The main part of the *UNIX* hand book, the part that explains the various *UNIX* commands, has an excellent index but there is none for *vi* and, although the hand book sections are detailed and fairly easy to understand, it is very difficult to find the particular activity you want to perform. There is however a useful detachable prompt card which covers the common commands and manoeuvres.

CONCLUSION

I have only glimpsed the full depth of capabilities of the text processing packages and indeed of the rest of *UNIX*. I anticipate it will take years to really get their full power at my fingertips; but what exciting years!. In next month's article I shall discuss some of the applications programs you can run on *UNIX*. Many have been designed to take the complexity and anxiety out of using this operating system.

It took me four or five days of intermittent use to learn as much about *UNIX* as is revealed in this article, and after the first day, I cannot say I found it too terrifying. As with all computer applications, learning your way around the hand book makes life much easier. If you find computers difficult, and have taken a long time to master the programs you are using at present, *UNIX* is not for you. However if you feel at home with present packages but are frustrated by their lack of power and limited capabilities, then you can buy *UNIX* with no hesitation. Finally if your personal productivity is of vital importance and you are prepared to devote money and the necessary time to mastering it (and remember that there are many books, tutorials and courses, as well as *Torch's* extremely helpful customer support unit to help in this) then you should have no hesitation. Like learning to fly a helicopter, it will mean that your ability to do what you want, quickly, efficiently, where you want and how you want, is revolutionized.



```
This is a UNIX terminal at work
This is on-screen underlining: it can
be switched on and then switched off!
This is on-screen italics which can
also be switched on an off easily
```

10. Underlining and italics can be shown on the screen.

Cheaper RAM

Mike Harrison

Sideways RAM on the BBC can be implemented in one of two ways: You can fit two RAM chips into a ROM board, if you have one, or you can buy a RAM-only unit. The first is fine if you already have a ROM board, as the chips are only about £5 each (Sept.). This article shows how these same chips can be used without a ROM board, i.e. in place of a RAM only board. The total cost is about £10 for the RAM chips, plus about 50p for the other chip used.

This cheap solution gives the same facilities as most ROM board based RAM, i.e. Auto-write (any write to memory in the ROM area goes to RAM, so data can be *LOADed into it directly, and machine code (e.g.ROM software) can be assembled directly to the RAM.), and the option to put on a 'Write protect' switch, to protect data stored in the RAM.

The key to the low cost, and ease of construction is the way that it is built: by 'piggy-backing' the two RAM chips, and plugging the bottom one into one of the BBC's ROM sockets.

GETTING TECHNICAL

Each of the RAM chips can hold 8K bytes of data, so two are needed to fill the 16K ROM area. External circuitry is needed to control which RAM chip is active, and to handle the writing of data to the RAM (and disabling of other ROMs when the RAM is written to). These functions are performed by a 74LS158 chip, which is a quad 2-input inverting multiplexer (don't worry if you don't know what that means, this isn't a tutorial on logic design, so just take it from me that it works!).

The circuit is shown in fig.1. IC3a selects which of the 2 RAM chips is accessed. IC1 is active from &A000 to &BFFF, and IC2 from &8000 to &9FFF. IC3b,c determine when the RAM is active – when the 'Chip Select' line from the socket is low, or the Read/Write line is low AND the 'Area Select' line from S21 is low. IC3d disables the ROMs when a Write occurs. The 'Read/Write' line is taken from pin 1 of IC72, as this is right next to the ROM.

Ultra cheap sideways RAM.

Fig. 1

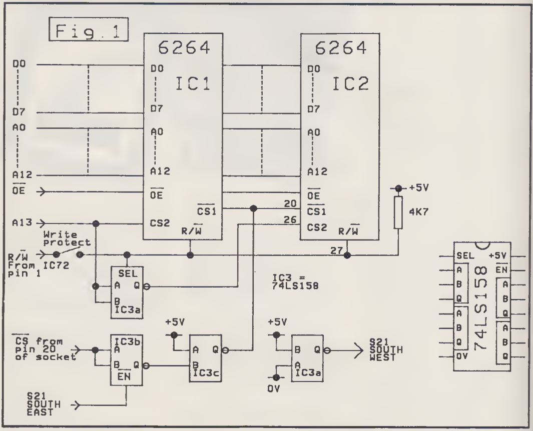
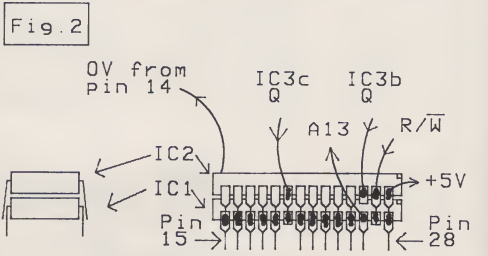


Fig. 2



A & B CLASSIFIED

Lineage: 40p per word.
Semi display: £9.00 per single column centimetre
Ring for information on series bookings/discounts.

All advertisements in this section must be prepaid
Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request)

01 - 437 0699
EXT 289

Send your requirements to:
CAROLINE FAULKNER, ASP LTD,
1 GOLDEN SQUARE, LONDON W1.

SOFTWARE ACCESSORIES

TURNKEY is a fully assembled M/C system to make all your datafile program disks secure with password protection. In unlocked state disks act normally (full catalogue and full information) but when locked can neither be read from nor written to. Simple Autoboot operation. **Send £12.95 plue S letter pessword, Vase, 1587, Gt. Weetern Road, Glaegow. Pleaee etete 40/80 tracke.**

Disc storage boxes, anti-static lockable with smoked glass cover, 2 sizes available:-

To store 70 disks	**£17.95**
To store 100 disks	**£19.95**

ED40, 46 The Maltings,
Stanstead Abbott,
Ware, Herts.

DISC BOXES £64.00 + VAT
ROM BOXES £49.99 + VAT

BBC Peripherals or the Amstrad education scheme and the Amstrad network.

contact:
N COMPUTERS LTD.,
Churchfield Road, Frodsham,
Cheshire WA6 6RD
Tel: (0928) 35700 (10 lines)

BBC/Flechertechnik Robot Interface

4 Outputs for motors, lamps, electro-magnets (MitoM4)
8 Inputs for the connection of switches, relays (E1 to E8)
2 Analogue inputs for potentiometers, photo resistors

Interfece only	**£39.95 + £1.50 p&p**
Interfece + kit 30554	**£99.95+ £2.50 p&p**

Red Giant Software Ltd
3A Oekcroft Close, Pinner, Middx
HA5 3YY

BBC/ELECTRON "IMAGE 2"

Without doubt the best back-up copier money can buy. Copes with probably all software. Send SAE for details of incredible features. Unlocks & locks progs. 100% M/C £4.80

PETER DONN (Dept AB)
18 Tyrone Road, Thorpe Bay, Southend-on-Sea, Essex SS1 3HS

ASTROLOGY

ASTROLOGY

for beginners. Special starter pack for only £11.50. Consists of a simple programme to calculate a horoscope, an introductory booklet and 2 self-teaching programmes (how to interpret the horoscope).

NO PREVIOUS KNOWLEDGE REQUIRED

for the B.B.C., A&B, Electron and many other micros, also wide range of programmes for more **Experienced Astrologers**. Cash with order (add 50p outside U.K.) or large SAE for free Catalogue to:

ASTROCALC (Dept A&B),
67 Peascroft Road, Hemel Hempstead, Herts HP3 8ER.
Tel: (0442) 51809.

EPROMS

BANK MANAGER

BBC Home Accounts, Random access disk-files, standing orders, budget codes, up to 10 accounts. Balance Sheets, spending graphs £10 add 30p P&P.

Paul Gardiner
103 Elder Close, Bedger Farm,
Winchester, Hants. SO22 4LL.
Tel: 0962 51443 eveninge.

SOFTWARE LIBRARIES

ELECTRON OWNERS We are a software library devoted to software for the Electron Computer. We have at present 80+ titles and we are continually updating with many of the top-selling games. Software lists should soon be well over **100.** Hire rates from 75p per fortnight. Life membership £6.00 (this includes free hire of first program). For further details please write to: Software Listing, Elderhurst, Lobleyhill Road, Meadowfield, Durham DH7 8RQ.

VIDEO

XXX VIDEO

Telephone (day or night)
0373 - 830563
FOR DETAILS

Glossy list from: **XXX Video Club, 2 Streatham High Rd., London SW16**
(You must be over 18 to ask for details)

SOFTWARE LIBRARIES

STAR-SOFT

(BBC + ELECTRON SOFTWARE LIBRARY)
Why buy programs when you can hire at a fraction of the cost?
JOIN THE BBC AND ELECTRON SOFTWARE LIBRARY
★★ Large varied selection of software including best selling titles ★★
★★ Expanding Catalogue ★★
★★ Hire rate from 75p per fortnight ★★
We offer a fast, quality service at competitive rates. Only £6 life membership inc first program free.
Send cheque/PO for £6 payable to **STAR-SOFT** or send stamp for catalogue: (please state whether BBC or Electron)
STAR-SOFT
16, Martland Cresc, Beech Hill, Wigan, Lancs. Tel: 0942 47574

FOR SALE

Printers:

CP-80 80cps Fric & Trac	**189.95**
CPA-00 100cps Fric & Trac	**199.95**
BBC Printer Cable 2 mtrs	**7.95**
3" HITACHI Disc drive cased complete	**79.95**
3" double sided discs for 2 **8.95** 4	**42.95**
Cassettes C15's boxed with label & card 3 **1.15** 10 **3.20** 100 **30.00**	
DISC PAYROLL Contracted in or out (Weekly, 2 or 4 Weekly & Monthly)	**59.45**
DISC ACCOUNTS Double entry Cashbook	36.45
1DISC RANDOM Access Mailing Full Sort	36.45

PRESTEL *2582020

Prices Include VAT. Postage & Packing add 50p

Micro-Aid
(BBC)
25 Fore Street, Praze, Camborne, Cornwall.
or 0209-831274 with ACCESS

Knights Electronics
(Computers & Music)
Computere & Peripherale for the serioue bualness & hobbylet

Our service does not stop at just giving competitive prices and a a fast friendly service.
Special Acorn, Atari 520ST & Amstrad Support
Sections own/other products

BBC 'B'	£277.00
BBC 'B' + DFS	£335.00
BBC 'B Plus'	£395.00
Acorn Muaic 500	£150.00
5.25" 5¼" Drives 40/80T from	£90.00
Kaga Vision 1	£199.00
Microvitec 1431	£165.00
Phillipa 7502 Green	£65.00
Phillips 2007 Col. TV/Mon	£186.00
Canon PW1080A NLQ	£255.00
CCP40 Colour Plotter	£70.00
Epson LX80 NLQ	£197.00
Juki 6000	£195.00
Juki 8100	£295.00
Kaga 810 NLQ	£229.00
MP185 NLO	£219.00

PLEASE PHONE IF YOU CANNOT SEE EXACTLY WHAT YOU NEED
All prices exclude carriage & VAT. We carry most leading brand names. Please ring to confirm latest prices. You will find us unbeatable For more information on how to get our regular discount price list & details of our other services, contact:

KECM 8 Westwood Lane Welling Kent DA16 2HE Telephone: 01-301-3745
(10am-10pm)
Callers by prior appointment only

FOR SALE. BBC Model B, plus disk drive, games and accessories. Phone Brewood 850959 after 6.00 pm.

BINDERS

ADVANCED/USER GUIDE BINDERS stiff plastic backed, open flat, titled ring binders £3.50 + £1.00 P&P. Hepworth & Co (Dept AB), Waulkmill Farm, Ingersley Vale, Bollington, Cheshire SK1D 5BP.

COURSES

RESIDENTIAL COURSES programming techniques in BASIC. Learn to use your computer on a weekend/holiday course. Details from:- **Jaysoft Microdevelopments, 2 Wester Row, Greenlaw, Berwickshire.**

SERIOUS SOFTWARE

FOR THE B.B.C. MICRO

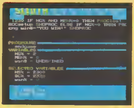
TOOLKIT

£27.00

- Adds 27 new commands for Basic program development and debugging
- Automatic error detection
- Full screen editor for Basic
- Host of utilities instantly available

SLEUTH

£29.00



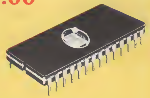
```
SLEUTH
 1230 IF NG% AND MEN%=0 THEN PROClost
ROCsetup ENDPROC ELSE IF NG%=0 THEN PR
ong win$="YOU WIN!" ENDPROC

PROCEDURE
  endgame
VARIABLES
  NG% = 2
  MEN% = 0
  win$ = UNDEFINED
SELECTED VARIABLES
  HS% = 2300
  SC% = 2300
  wink = 0
```

- Revolutionary new debugging tool for Basic programmers
- Dual screen operation
- Single step facility
- Speed adjuster
- Gives YOU control over Basic.

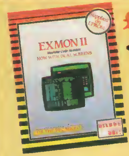
HELP

£25.00



- The Information ROM
- On screen Help facility to aid programming
- Includes Basic keywords • Sound • Colour • Teletext • VDU commands • FX Calls • GCOL • Screen Modes • Memory usage • PAGE/HIMEM etc

EXMON II



£29.00

- Dual screen operation
- Full screen memory editor
- 60 machine code commands
- Invaluable if you wish to program, debug, or learn about assembler

!FREE!

information pack

To find out about these and other Beebugsoft products please phone for a FREE information pack. 24 hour 0494 81 5228

BEEBUG SOFT

DISCMASTER

£19.00

- Eleven powerful utilities to support the disc user:- Disc Editor • String Search • Recover • Movedown • Disc Menu • Indexer • Copier • Link • Overlay • 40 to 80 • Dual Catalogue • Formatter

MASTERFILE II

£19.00

- General purpose file management package
- Allows large amounts of information to be stored and processed
- Extremely flexible
- Easy to use

ORDERING INFORMATION

All these products are available from your local dealer and larger branches of Boots.

Post and packing 50p Prices include VAT

Telephone orders (24 hours) 0494 81 6666

Access

Mail order:- P.O. Box 50 St Albans Herts

For further information telephone (0727) 60263

SPELLCHECK II

£31.00

- ROM-based spelling checker for Wordwise or View
- Extremely fast on test timings Watford 1.3 DFS – 920 wpm Acorn 1.2 DFS + tube – 1450 wpm
- 6,000 word dictionary disc
- Dictionary increased with usage